Noname manuscript No.
(will be inserted by the editor)

Multi-Cue Contingency Detection

Jinhan Lee · Crystal Chao · Aaron F. Bobick · Andrea L. Thomaz

Received: date / Accepted: date

Abstract The ability to detect a human's contingent response is an essential skill for a social robot attempting to engage new interaction partners or maintain ongoing turn-taking interactions. Prior work on contingency detection focuses on single cues from isolated channels, such as changes in gaze, motion, or sound. We propose a framework that integrates *multiple cues* for detecting contingency from multimodal sensor data in human-robot interaction scenarios. We describe three levels of integration and discuss our method for performing sensor fusion at each of these levels. We perform a Wizard-of-Oz data collection experiment in a turn-taking scenario in which our humanoid robot plays the turn-taking imitation game "Simon says" with human partners. Using this data set, which includes motion and body pose cues from a depth and color image and audio cues from a microphone, we evaluate our contingency detection module with the proposed integration mechanisms and show gains in accuracy of our multi-cue approach over single-cue contingency detection. We show the importance of selecting the appropriate level of cue integration as well as the implications of varying the referent event parameter.

ONR YIP N000140810842.

Jinhan Lee
801 Atlantic Drive Atlanta, GA 30332-0280
School of Interactive Computing
E-mail: jinhlee@cc.gatech.edu

Crystal Chao
E-mail: cchao@cc.gatech.edu

Aaron F. Bobick
E-mail: afb@cc.gatech.edu

Andrea L. Thomaz
E-mail: athomaz@cc.gatech.edu

Keywords Contingency Detection · Response Detection · Cue Integration

1 Introduction

A social robot can use contingency as a primary or auxiliary cue to decide how to act appropriately in various interaction scenarios (e.g. [1,2]). One such scenario is identifying willing interaction partners and subsequently initiating interactions with them, such as a shop robot's attempting to engage customers who might need help. To do this, the robot would generate a signal intended to elicit a behavioral response and then look to see if any such response occurs. The presence of a change in the behavior of a human at the right time is a good indication that he or she is willing to follow-up with an interaction. Similarly for disengagement, the absence of contingent responses can verify that a human has finished interacting with the robot.

Another scenario is that which involves reciprocal turn-taking situations. In such contexts, contingency can be used as a signal that helps a robot determine when the human is ready to take a turn and when it is appropriate for the robot to relinquish the floor to the human. For example, if the robot sends a signal to yield the floor to the human and detects a contingent response from the human mid-way through, this may indicate the human's readiness take a turn. In this case, the robot can interrupt its signal and yield the floor to the human.

When a robot understands the semantics of a human's activity in a given context and has a specific expectation over a set of known possible actions, then checking for a contingent response might simply entail matching the human's executed action against the ex-

pected action set. This strategy of *detecting expected reactions* makes the sometimes problematic assumptions that the set of appropriate or meaningful responses can be enumerated in advance, and that specific recognition methods can be constructed for each response. For example, the robot might expect the human to utter a sentence that is contained in the grammar of its speech recognizer; a match would indicate a contingent response. In these cases, preprogrammed domain-specific responses may be sufficient to decide how to act. In many realistic interaction scenarios, however, the potential natural human responses are so varied that it would be difficult to enumerate them a priori and to preprogram methods of detecting those actions. In addition, knowing which action was generated is not always necessary; often, the presence of a contingent action is enough to enable appropriate social behavior. Thus, a robot needs an additional mechanism to detect contingent responses that are not modeled in advance: *detecting a behavioral change* that is indicative of a contingent response. We believe that these two mechanisms are complementary for contingency detection. Our main focus in this work is to provide a framework that detects behavior changes without a priori knowledge of possible responses.

A contingent behavioral change by a human can occur in one or multiple communication channels. For example, a robot that waves to a human to get his attention may receive the speech of "Hello!" with a simultaneous wave motion as a response. Here, we consider the problem of human contingency detection with multimodal sensor data as input when forming our computational model. We validate the multiple-cue approach using multimodal data from a turn-taking scenario and show that modeling a response using multiple cues and merging them at the appropriate levels leads to improvements in accuracy in contingency detection. Collectively with our previous work [3], the results in this paper demonstrate that our behavior-change-based contingency detector provides a highly indicative perceptual signal for response detection in both engagement and turn-taking scenarios.

In this paper, we make the following contributions.

1. We present a contingency detection framework that integrates multiple cues, each of which models different aspects of a human's behavior. We extend our prior work, which only uses the visual cues of motion and body pose, to create a more generic contingency detection module. In our proposed framework, the cue response can be modeled as either an event or a change.
2. We propose three different levels of cue integration: the frame level, the module level, and the decision level. We show that for change-based detection, integration of visual cues at the module level outperforms integration at the decision level.
3. We examine the effects of selecting different timing models and referent events. In particular we show how selecting the *minimum necessary information* referent event [4] improves detection and requires a smaller amount of data, increasing the tractability of the real-time detection problem.
4. We provide a probabilistic method for measuring the reliability of visual cues and adaptively integrating those cues based on their reliability.
5. We evaluate our proposed contingency detection framework using multimodal data and demonstrate that multi-cue contingency detection is a necessary component for interactions with humans and their multimodal responses.

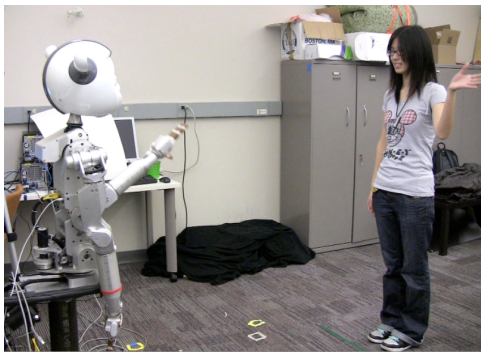


Fig. 1 A human being contingent to the robot in a turn-taking scenario based on the game "Simon says." The robot sends both motion and speech signals to the human subject by simultaneously waving and saying hello. The subject waves back to the robot in response.

The rest of paper is organized as follows. Section 2 shows prior research related to contingency detection. Section 3 describes our computational model for contingency detection and cue integration. Section 4 introduces two different approaches for response detection: event-based detection and change-based detection. Section 5 describes the derivation of motion, body pose, and audio cues from image and depth sensors and the implementation of contingency detectors using these cues. Section 6 explains the different levels of cue integration in detail. Sections 7–10 explain human data collection, model evaluation with the collected data, and experimental results. Finally, Section 11 presents our conclusions.

2 Related Work

Prior work has shown how contingency detection can be leveraged by a social robot to learn about structure in

functional or social environments. Contingency detection has been used to allow robots to learn human gaze behaviors [5–7] and to understand social interactions [1, 2, 8].

Butko and Movellan [7] broke down the problem of contingency detection into two subproblems: response detection and timing interpretation. They developed a model of behavior that queried the environment with an auditory signal and detected a responsive agent using a thresholded sound sensor. The goal of the controller was to query at times that maximized the information gained about the responsiveness of the agent. Thus, their focus was on the timing constraints of the contingency problem. Similarly, Gold and Scassellati [9] focused on learning effective timing windows for contingency detection, which they did separately for an auditory signal and a visual signal.

There is evidence that humans use this contingency mechanism to learn some causal relationships. Watson found that infants use this mechanism for self-recognition and for detection of responsive social agents [10, 11]. A robot can also use contingency for recognition of self and others [12, 13]. In this formulation of the problem, the presence of a visual change is directly mapped to the presence of a response. Thus, the presence of the response is assumed, but the source of the response is unknown and is attributed using timing interpretation.

In our work, we formulate the problem slightly differently. Because we are interested in human-robot interaction domains, such as engagement detection and turn-taking, we focus on a single response source: the human. We also cannot assume that visual changes always indicate responses, as visual changes frequently occur without correlating with contingent responses. Detecting contingency thus requires more complex analysis of what visual changes are observed. Previously in [3], we implemented contingency detection using single vision-based cues and demonstrated the application of our contingency detection module as a perceptual component for engagement detection. Other work has focused on processing other individual channels, independently demonstrating the significance of gaze shift [14, 15], agent trajectory [16, 17], or audio cues [18] as contingent reactions.

An ideal contingency detector should be able to accept a variety of sensory cues, because certain perceptual and social cues are more informative for some interactional situations than for others. Here we extend the modalities supported by our framework in [3] to include the audio channel and detail an approach for the integration of multiple cues.

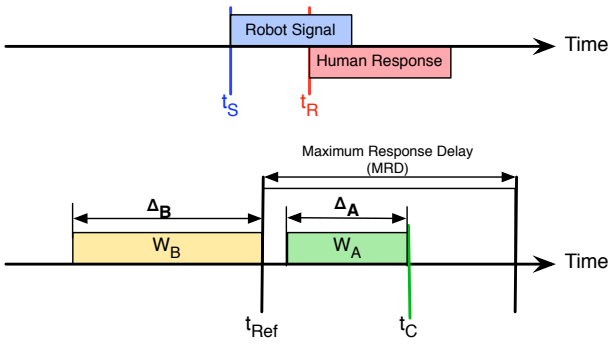


Fig. 2 Causal relationship between a robot's interactive signal and a human's response. Top: t_S is the start of the robot signal, and t_R is the start of the human response. Bottom: Two time windows, W_B and W_A, defined with respect to the current time t_C, are used to model the human behavior before and after t_{Ref}. W_B and W_A are of size Δ_B and Δ_A respectively. The time window starting at t_{Ref} and valid over MRD is examined for the contingent human response. Note that t_{Ref} may not be same as t_S.

3 Approach

Contingency detection consists of two sub-problems: response detection and timing estimation. Figure 2 shows the causal relationship between a robot signal and the corresponding human response as well as time windows for detecting such a response. A robot generates some interactive signal to a human by gesturing to, speaking to, or approaching her. Some time after the robot initiates a signal and indeed sometimes before the robot completes its action, the human may initiate a response that could last for a certain duration.

The time interval during which a human may respond to a given robot signal needs to be estimated. We define its endpoints to be the time at which the robot begins to look for a human response, t_{Ref}, and the maximum time for the human's response delay after t_{Ref}, $(t_{Ref} + MRD)$. To detect a response, we evaluate sensor data within that time window. Because we are not considering anticipation on the part of the human, t_{Ref} is defined to be after t_S; to be effective it also should precede the initiation time of the human's response, t_R. Chao et al. [4] define the notion of the *minimum necessary information*(MNI) moment in time when an actor - human or robot - has conveyed sufficient information such that the other actor may respond appropriately. We will describe this concept in more detail within the context of our experiments in Section 7.) and in the results we will show the MNI is a more effective referent than the time at which the robot completes its signal.

To detect human responses within the estimated evaluation time window, we take two different modeling approaches: event detection and change detection. As

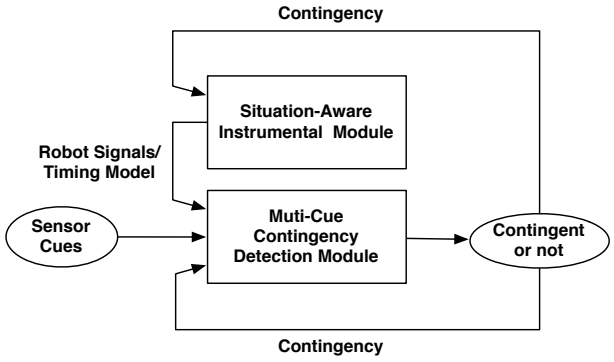


Fig. 3 Situation-independent Contingency Detection. Situation specific information, such as robot signals and timing models are parameters of the contingency module.

shown in Figure 2, we define W_B and W_A as time windows from which data are used to model the human behavior before and after the robot's referent signal, t_{Ref}, respectively. An event detection models a response as an event and that event is looked for in W_A. On the other hand, a change detection models a response as a change, which is measured by comparing observations between W_B and W_A. When the human does not respond to the robot, both W_B and W_A describe the same aspect of the human's behavior. However, in the contingent cases the human changes her behavior to make a contingent response, thus W_B and W_A model different behaviors. To detect such changes, we measure how likely that sensor data in W_A reflects the same behavior observed in the sensor data in W_B.

Figure 3 shows information flow between a contingency module and other robot modules. To make our contingency detection module as situation-independent as possible, we assume that the expected timing of a response is determined by another of the robot's modules and is taken as a parameter to the contingency detection module.

3.1 Cue Representation

To model a given aspect of human behavior, we derive information from observations from a single or multiple sensors, hereafter referred to as a *cue*. Depending on the characteristics of the sensors used, a cue is encoded as either a binary variable or a continuous variable. When underlying sensors produce low-dimensional observations, the derived cue is easily represented as a binary variable. For example, audio and touch sensor signals can be classified as either on or off by simply thresholding the magnitude of the raw observations. Sensors that generate high-dimensional observations, such as image and depth sensors, require more complicated preprocessing, and thus a derived cue would be encoded as continuous and high-dimensional variables. Section 5 describes procedures for extracting cues from sensors in detail.

3.2 Multi-cue Integration

The extracted cues from sensors should be integrated in such a way that the contingency detection module reduces uncertainty and increases accuracy in its decision-making. We adapt the data fusion framework introduced by Hall and Llinas [19] to integrate cues. Here, we define a frame as a representation of cue information and as an input to the contingency detection module. We define three levels of integration: 1) the *frame level*, at which cues are merged into one augmented frame as an input to the contingency module; 2) the *module level*, at which cues are integrated within one contingency detection module; and 3) the *decision level*, at which outputs from multiple single-cue contingency modules merge. These levels are shown in Figure 4.

Intuitively, the principal difference between frame and module level integration and, decision level integration is whether the cues are combined into a single signal whose variation during the contingency window is evaluated for a behavioral change, or whether each cue is considered independently and the two decisions are fused to provide a final answer.

Cues should be integrated at the right level based on characteristics of a cue, the underlying sensor's sampling rate and dimensionality, and encoded perceptual information. If two cues are encoding the same perceptual modality of a human and they complement each other, thus modeling behavior in a more discriminative way, then two cues should be merged either at the frame or at the module level rather than at the decision level.

The difference between frame level integration and module level integration is whether the cues are augmented and evaluated by one common distance metric, or whether each individual cue is evaluated by a cue-specific distance metric and the two evaluations are fused. One evaluation on merged cues captures correlation between cue observations better than merging multiple individual evaluations. If such a distance metric is available, cues should be integrated at the frame level rather than at the module level. Because of the dissimilarity of the physical measurements we do not explore frame level integration in this paper.

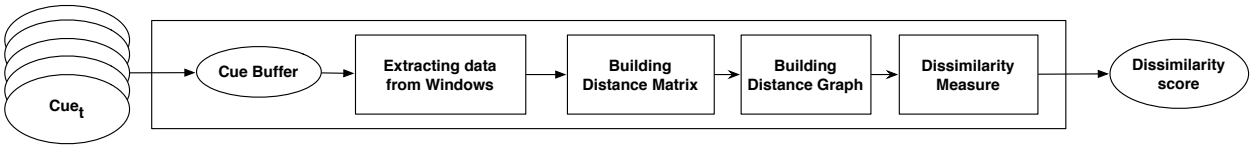


Fig. 5 Change Detection Framework [3]

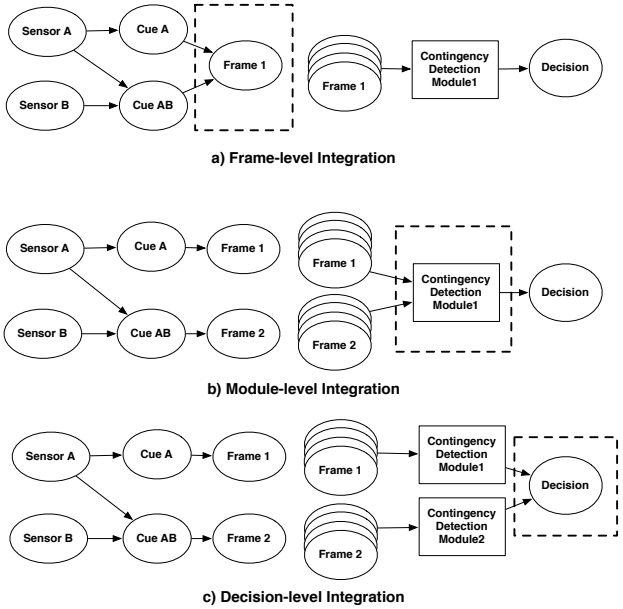


Fig. 4 Cue integration at different levels: a) at the frame level, b) at the module level, and c) at the decision level.

4 Response Detection

4.1 Event Detection

When the possible forms of expected responses are known to a robot a priori, responses can be modeled and be recognized as events. If temporal events are modeled with a high-dimensional cue, some vision-based methods such as generative models (Hidden Markov Models and their variants) and discriminative models (Conditional Random Fields and their variants) can be adopted [20]. Simple events such as touching a robot or making a sound can be modeled using low-dimensional cues, so they are easier to detect using simple filtering methods. To detect vocal responses, we derive the audio cue from a microphone sensor and model them as events.

4.2 Change Detection

Change detection measures a human's behavioral change as a response to a robot's signal. We look for a significant perturbation in the human behavior by modeling that behavior before and after the signal and looking for significant differences. We assume that any observed perturbation happening within the allowed time interval is the human's response. A change is measured using a history of cue data between W_B and W_A as input. To measure the degree of behavioral difference in the response, we proposed a change detection framework [3], as shown in Figure 5.

As the change detector accumulates cue data into a buffer over time, it keeps only enough to fill the past Δ_B amount of time and discards frames that are older. When triggered at time t_{Ref}, the data for W_B stays fixed, and the detector transitions to accumulating cue data to fill W_A until time $t_{Ref}+\Delta_A$.

Once cue data exist over all of W_B and W_A, a cue distance matrix is calculated between data in W_B and W_A using a cue-specific distance metric (Section 4.2.1). The distance matrix is converted into a distance graph by applying specific node connectivity (Section 4.2.2). Then, we calculate the dissimilarity score by measuring a statistical difference between the graph nodes representing data from W_B and W_A (Section 4.2.3). Finally, as an extension to [3] and as one of the contributions of this paper, we introduce a method that uses probabilistic models to evaluate a dissimilarity score S within our multi-cue contingency detection framework (Section 4.2.4).

4.2.1 Building the Distance Matrix

We define a cue buffer to be the set of cue feature vectors v_t computed at each instant of time. From the cue buffer, we first extract cue data from two time windows W_B and W_A based on t_{Ref} and t_C. W_B is the time interval between t_{Ref} - Δ_B and t_{Ref}, and W_A is the time interval between t_C - Δ_A and t_C. Let V^B and V^A denote cue data extracted from the time windows W_B and W_A respectively:

$V^B = \{v_t \mid t \in W_B\} = \{v_{l+1}, v_{l+2}, ..., v_{l+P}\}$

$V^A = \{v_t \mid t \in W_A\} = \{v_{m+1}, v_{m+2}, ..., v_{m+Q}\}$.

Let $V = V^B \cup V^A$, so $|V| = (P+Q)$. Let V_i denote the i^{th} element of V. The distance matrix DM_X of a cue X is calculated by measuring the pairwise distance between cue elements in V; $DM_X(i,j)$ describes the distance between cue vectors V_i and V_j using a predefined distance metric for the cue X. We will describe

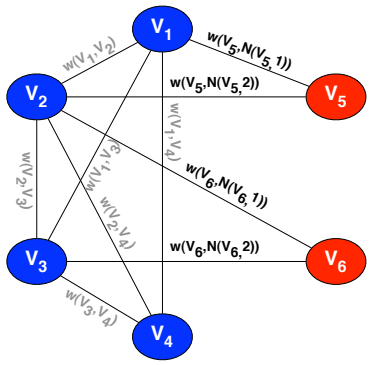


Fig. 6 Building a distance graph from the distance matrix. The edge weight between V_i and V_j, w(V_i, V_j), is the distance between V_i and V_j, $DM_X(i,j)$. N(V_i, k) is the k_{th} nearest neighbor of V_i in W_B. Blue and red nodes are from V^B and V^A, respectively.

distance metrics for visual cues, motion (see Section 5.2), and body pose (See Section 5.3).

4.2.2 Building the Distance Graph

We construct a distance graph from the distance matrix. The distance graph has the following characteristics as shown in Figure 6:

1. Nodes from W_B (e.g. V_1 to V_4) are fully connected to each other.
2. Nodes from W_A (e.g. V_5 and V_6) are never connected to each other.
3. Nodes from W_A are only connected to the κ nearest nodes from W_B.

The edge weight between two nodes V_i and V_j, $w(V_i, V_j)$, corresponds to $DM_X(i,j)$.

4.2.3 Calculating the Dissimilarity Measure

We measure dissimilarity by calculating the ratio of the cross-dissimilarity between V^B and V^A to the self-dissimilarity of V^B. $N(V_i, k)$ denotes the k_{th} nearest neighbor of V_i in V_B. Let E denote the number of dissimilarity evaluations. $CD(V)$ measures the cross-dissimilarity between V^B and V^A in the following equation:

$$CD(V) = \sum_{q=1}^{Q}\sum_{k=1}^{\kappa}\sum_{e=1}^{E} w(V_{P+q}, N(N(V_{P+q},k),e)), \quad (1)$$

where P is $|V^B|$ and Q is $|V^A|$.

$SD(V)$ measures the self-dissimilarity within M_T^B:

$$SD(V) = \sum_{q=1}^{Q}\sum_{k=1}^{\kappa}\sum_{e=1}^{E} w(N(V_{P+q},k),N(N(V_{P+q},k),e)) \quad (2)$$

The dissimilarity of V, $DS(V)$, is defined as:

$$DS(V) = \frac{CD(V)}{SD(V)} \quad (3)$$

$DS(V)$ is the dissimilarity score S for a given V.

4.2.4 Evaluating Dissimilarity Score

In [3], we learned a threshold value on the dissimilarity score from the training data and used that to classify the score as being contingent or not. This simple evaluation method cannot be used in our probabilistic model for multi-cue integration because it does not have the confidence on the decision made, and because it does not take into account how informative a used cue is.

We propose a new evaluation method that resolves the two problems described above. To determine that an observed change (i.e. a dissimilarity score) actually resulted from a human response and not from changes that occur naturally in the human's background behavior, we should evaluate the change not only under the contingency condition, but also under the non-contingency condition. To this end, we model two conditional probability distributions: a probability distribution of the dissimilarity score S under the contingency condition C, $P(S|C)$, and a probability distribution of S under the non-contingency condition, $P(S|\overline{C})$. Assuming that a human changes her behavior when responding, these two distributions need to differ for the cue to be considered informative.

We learn the distribution $P(S|C)$ off-line from training data in which human subjects are being contingent to the robot's action. We estimate the distribution $P(S|\overline{C})$, the null hypothesis, on the fly during an interaction from observations of the human's behavior before the robot triggers a signal. It is important to note that a null hypothesis is estimated with on-line data, particularly the data gathered immediately before the robot's signal is triggered. As shown in Figure 7, this distribution is estimated from dissimilarity score samples, each of which is obtained as if the robot's signal were triggered and enough data were accumulated at each point in time.

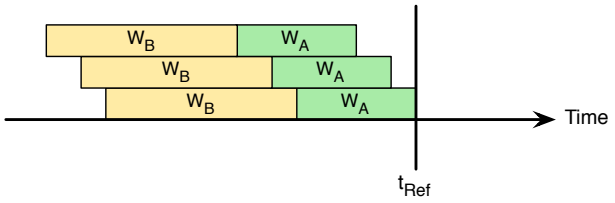


Fig. 7 A method for building null hypothesis. Dissimilarity score samples are obtained by evaluating data in windows over time.

5 Cues for Contingency Detection

The choice of cues for response detection should be determined by the nature of the interaction. When a robot engages in face-to-face interaction with a human, a shift in the human's eye gaze is often enough to determine the presence of a response [14,15]. In situations where a gaze cue is less indicative or is less reliable to perceive, other cues should be leveraged for response detection. Here, we are interested in modeling human behavior using three different cues: 1) a *motion cue*, the pattern of observed motion; 2) a *body pose cue*, the observed human body configuration; and 3) an *audio cue*, the presence of human sound.

5.1 Audio Cue

For the audio cue, the contingent response is modeled as an event that occurs during the evaluation time window W_A (from t_{Ref} to t_{Ref}+MRD). The audio cue models the presence of the human response through the sound channel and is derived from a microphone sensor. We build a simple event detector using the audio cue. The audio cue is estimated as a binary variable by thresholding on raw sound data to remove a base level of background noise, as shown in Figure 8(a). Only raw sound samples greater than a certain threshold are set to 1; otherwise, they are set to 0. At time t, this binary audio cue, denoted as A_t is added to the audio cue buffer. After a robot triggers a signal, the evaluation starts. During the evaluation, at time t_c, cue data A_T is first extracted from W_A: $A_T = \{A_t \mid t \in W_A\}$. A_T is classified as A_{on} only if all elements in A_T are 1; otherwise it is classified as A_{off}. The classification ratio of A_{on}, $R(A_{on})$, is calculated as:

$$R(A_{on}) = \frac{P(C|A_{on})}{P(\overline{C}|A_{on})} = \frac{P(A_{on}|C)P(C)}{P(A_{on}|\overline{C})P(\overline{C})} \quad (4)$$

Since the presence of a human response can only be considered when the audio cues are onset, the classification ratio of A_{off} is set to 1 and thus does not have any influence on the overall contingency decision when it is integrated with other cues.

5.2 Motion Cue

The motion cue models the motion patterns of the human of interest in the image coordinate system. This cue is derived from observations from image and depth sensors. To observe only motions generated by a human subject in a scene, we segment the region in which a human subject appears. To do so, we use the OpenNI API [21], which provides the functionalities of detecting and tracking multiple humans using depth images. The process for generating the motion cue is illustrated in 8(b). First, the motion in an image is estimated using a dense optical flow calculation [22]. After grouping motion regions using a connected components-based blob detection method, groups with small motions and groups that do not coincide with the location of the human of interest are filtered out. The motion cue comprises the remaining motion regions. This extracted motion cue is used as an input into the motion contingency module and is accumulated over time.

In the evaluation of [3], we included an intermediate step of dimensionality reduction using Non-negative Matrix Factorization (NMF) [23] on the combined cue data. The NMF coefficients of the data are normalized, so the magnitude of the original motion is encoded in NMF basis vectors. We calculate Euclidean distances between the cue data using only NMF coefficients of the data. When the human does not move, small motions that occur from noise observations can stand out and determine a dissimilarity score. If some noise occurs in M_T^A, then a high dissimilarity score would be produced. To remove this instability, we now reconstruct them using NMF basis vectors to recover the original magnitude of motion and calculate Euclidean distances between the reconstructed cue data. The small value ϵ_M is added to the distance matrix of M_T^B to increase the overall self-dissimilarity of M_T^B.

5.3 Body Pose Cue

The body pose cue models the body configuration of the human of interest. The human body configuration is estimated from a 3D point cloud extracted from the human of interest. The process for generating the body pose cue is illustrated in 8(c). As for the motion cue, we segment the region of the human from a depth scene

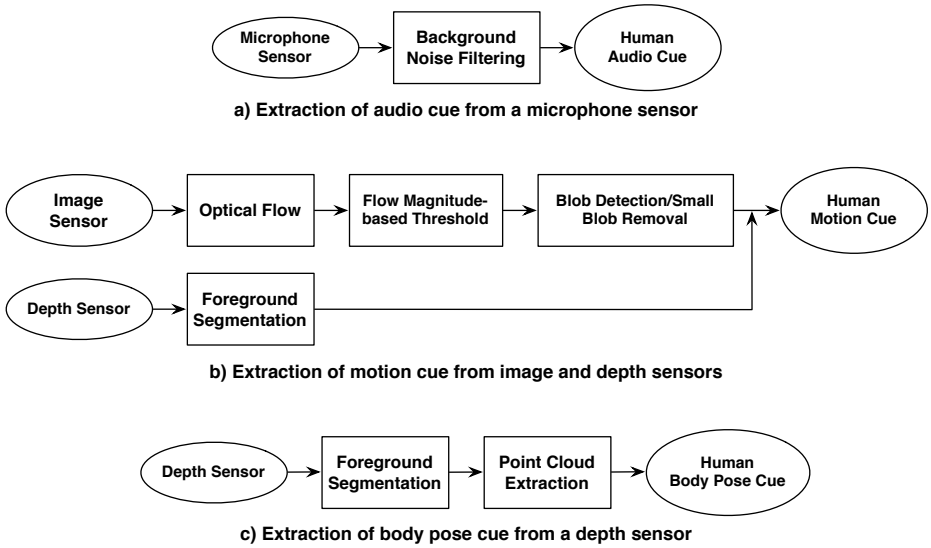


Fig. 8 Extracting cues from different sensors: a) audio cue, b) motion cue, and c) body pose cue.

using the OpenNI API. Then, we extract a set of 3D points by sparse sampling from the human region in a depth image and reconstructing 3D points from those depth samples.

As a distance metric for the body pose cue, we measure the pairwise distance between two sets of 3D points. Let P_1 and P_2 be two sets of 3D points: $P_1 = \left\{X_i^1 \mid i = 1, \ldots, m\right\}$ and $P_2 = \left\{X_j^2 \mid j = 1, \ldots, n\right\}$. The distance between these point sets P_1 and P_2, $PD(P_1, P_2)$, is defined as follows:

$$PD(P_1, P_2) = \frac{1}{mK} \sum_{i=1}^{m} \sum_{k=1}^{K} \left\| X_i^1 - X_{ik}^2 \right\|_2 + \frac{1}{nK} \sum_{j=1}^{n} \sum_{k=1}^{K} \left\| X_j^2 - X_{jk}^1 \right\|_2 , \quad (5)$$

where X_{ik}^2 is the k_{th} nearest neighbor in P_2 to X_i^1 and X_{jk}^1 the k_{th} nearest neighbor in P_1 to X_j^2. When calculating the distance matrix DM_D of M_T^B and M_T^A, the small value ϵ_D is added to the distance matrix of M_T^B for the same reason of handling noise effects as for the motion cue.

6 Multiple Cue Integration and Adaptation

Cues can be integrated at one of three levels: the frame level, the module level, or the decision level. The specific level should be chosen based on characteristics of the cues, such as dimensionality, sampling rates, and associated semantics.

Since the motion and body pose cues are extracted at similar sampling rates from image and depth sensors and also model the same perceptual region, we can combine the motion and configuration aspects to yield a feature that is more discriminative than either separately. Because a common distance metric for these cues is not available, we merge these cues at the module level, at which distance matrices calculated from individual cues are merged. Due to the different sampling rate and low-dimensional signal, the audio cue is integrated at the decision level with the output of the merged motion and body pose cues. For the decision level of integration, we use a naïve Bayes probabilistic model. Our implementation of the proposed framework for contingency detection with the motion, body pose, and audio cues is shown in Figure 9.

6.1 Cue Integration at the Decision Level

At the decision level of integration, we use the naïve Bayes probabilistic model to integrate dissimilarity scores obtained from cues. We chose this model because cues that are integrated at this level are assumed to be conditionally independent of each given the contingency value; otherwise, they should be integrated at other levels. Assume that some function f_X is provided that summarizes the current and past cue observations into a dissimilarity score. For two cues Cue_i and Cue_j that are integrated at the decision level, the overall contingency C in terms of $S_i = f_i(Cue_i)$ and $S_j = f_j(Cue_j)$ is estimated by standard Bayesian mechanics, as de-

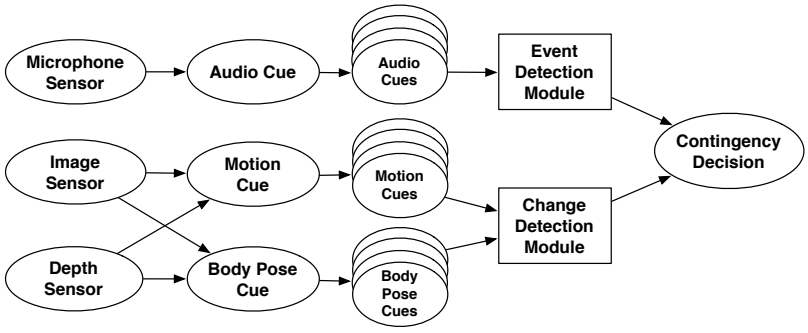


Fig. 9 An implementation of the proposed contingency framework with the motion, body pose, and audio cue.

scribed in Equation 8:

$$P(C|S_i,S_j) = \frac{P(S_i,S_j|C)P(C)}{P(S_i,S_j)} = \frac{P(S_i|C)P(S_j|C)P(C)}{P(S_i,S_j)} \quad (6)$$

$$P(\overline{C}|S_i,S_j) = \frac{P(S_i,S_j|\overline{C})P(\overline{C})}{P(S_i,S_j)} = \frac{P(S_i|\overline{C})P(S_j|\overline{C})P(\overline{C})}{P(S_i,S_j)} \quad (7)$$

$$\frac{P(C|S_i,S_j)}{P(\overline{C}|S_i,S_j)} = \frac{P(S_i|C)}{P(S_i|\overline{C})}\frac{P(S_j|C)}{P(S_j|\overline{C})}\frac{P(C)}{P(\overline{C})} \quad (8)$$

If this ratio > 1, we declare that the human behavior is changed; otherwise, any changes are not detected.

6.2 Integrating Motion and Body Pose Cues at the Module Level

At the module level of integration, the cue data are integrated when calculating a distance matrix. From the accumulated cue data for both motion and body pose cues, two distance matrices are calculated independently and merged into a new distance matrix. Since the motion and body pose cues are represented in different coordinate spaces, an image space for motion and a 3D world space for body pose, the distance matrices need to be normalized. We denote the distance matrices for motion and body pose cues as DM_M and DM_D, respectively. The merged distance matrix DM_N is calculated in the following way:

$$DM_N = \frac{1}{2}\left(\frac{DM_M}{\|DM_M\|_F} + \frac{DM_D}{\|DM_D\|_F}\right), \quad (9)$$

where $\|DM_X\|_F$ is the Frobenius norm of a matrix DM_X. After building DM_N, we use this distance matrix to calculate the dissimilarity measure, as explained in Section 4.2.3.

6.3 Audio Cue Integration

Because of the low dimensionality and fast sampling rates of the audio cue, the audio is naturally integrated with other cues at the decision level. We model the specific event of sound being on for some amount of time. In contrast to the motion and body pose cues, probabilistic distributions of this event for the naïve Bayes model are learned off-line from the training data.

7 Data Collection

Two important scenarios in HRI where contingency detection is applicable are engagement detection and turn-taking. In previous work, we demonstrated our contingency detection module with single cues in an engagement detection scenario using Simon, our upper-torso humanoid robot [3]. In that experiment, the robot attempted to get a human subject's attention while the human was performing a background task, such as playing with toys or talking on a cell phone.

In this paper, we validate multiple-cue contingency detection within a turn-taking scenario in which the robot plays an imitation game with a human partner. Importantly, this imitation game was designed not for evaluation of a contingency detector but for generating natural interactions with human-like timings. To collect naturalistic data, the robot was tele-operated over the whole interaction by one of authors with randomly generated timing variations. In this section, we describe how we conducted data collection for this turn-taking scenario.

This imitation game was based on the traditional children's game "Simon says," and is described extensively in [4]. The interaction setup is shown in Figure 1. In this game, one participant plays the leader and the other plays the follower. The leader is referred to as "Simon." There are two phases in the interaction, a

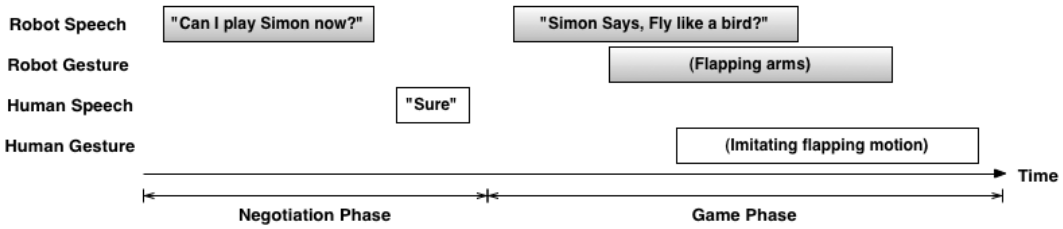
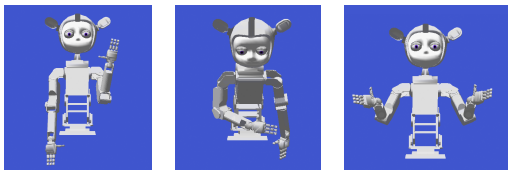
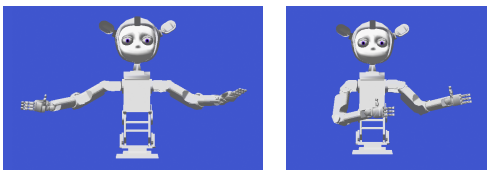


Fig. 10 An example of an interaction between a human and a robot.



(a) "Wave" (b) "Bow" (c) "Shrug"



(d) "Fly like a bird" (e) "Play air guitar"

Fig. 11 Actions in the "Simon says" game.

game phase and a negotiation phase. An example of an interaction is shown in Figure 10.

During the game phase, the leader says sentences of the structure, "Simon says, [perform an action]" or simply "[Perform an action]." The five motions understandable to the robot were waving, bowing, shrugging, flying like a bird, and playing air guitar. These motions are shown in Figure 11. The follower must imitate the leader's action when the leader starts with "Simon says," but when the leader does not, the follower must refrain from performing the action. If the follower mistakenly performs the action when the leader does not start with "Simon says," the follower loses the game. The leader concludes a game phase segment after observing an incorrect response by declaring, "You lose!" or "I win!"

In the negotiation phase, the leader and the follower negotiate about switching roles. The follower can ask, "Can I play Simon?" or state, "I want to play Simon." The leader can acquiesce to or reject the follower's request. The leader also has the option of asking the follower, "Do you want to play Simon?" or saying to him, "You can play Simon now." Similarly, the follower can acquiesce to or reject the leader's request. The leader and follower can exchange roles at any time during the interaction.

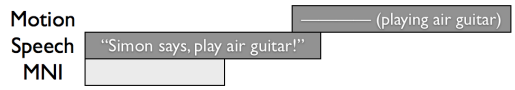
We collected multimodal data from 11 human subjects. There were about 4 minutes of data per subject. The sensors recorded were one of the robot's eye cameras, an external camera mounted on a tripod, a structured light depth sensor ("Kinect") mounted on a tripod, and a microphone worn around the participant's neck. We also logged the robot's signals with timestamps, so that we could link the time of the robot's signal with the sensor data. The robot's eye camera and the external camera were intended to extract a human subject's face information, which we did not use in this paper. We used only depth and camera data from Kinect sensor and microphone sensor data.

8 Model Building and Evaluation

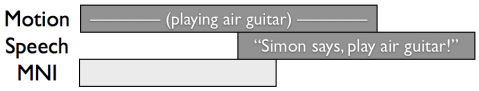
To build and evaluate a computational model of contingency detection for the "Simon says" interaction game, we use a supervised learning approach. We employ a leave-one-subject-out cross validation procedure; a single subject is iteratively left out of training data and is used as testing data.

First we split the recorded data sessions into shorter video segments, each of which starts at the t_{Ref} of one robot signal and ends at the t_{Ref} of the following one. We examine two different points of time as a referent event, t_{Ref}. One referent point is t_S, the time at which the robot signal starts, and the other is the point of minimum necessary information (MNI), the time at which the robot has finished conveying enough information for a human to give a semantically appropriate response.

The time location of the MNI was coded from video by two authors. In the game phase, the human needs to know whether or not to respond and also which motion to respond with if she needs to respond, so the MNI is the point at which both pieces of information have been delivered. In the negotiation phase, since the main channel for delivering information is speech, the information is sent by a pronoun. Some examples of coding robot MNI in the game phase and the negotiation phase are shown in Figure 12 and Figure 13, respectively. The coder agreement was 99.8% for robot MNI events.

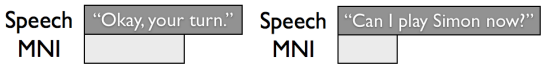


(a) All informative speech occurs before the animation starts.



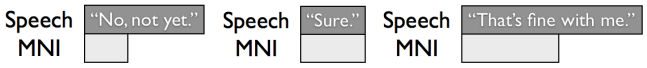
(b) The action is conveyed through motion before the human knows whether or not to execute it.

Fig. 12 Examples of coding robot MNI in the game phase.



(a) Pronouns demarcate information for turn exchange.

(b) The emotive phrase announces the end of a set.



(c) Examples of acknowledgments.

Fig. 13 Examples of coding robot MNI in the negotiation phase.

Depending on the presence or absence of a human's response, video segments were partitioned into two sets, contingent and non-contingent. In both phases, if the human makes noticeable body movements or vocal utterances as a response to a robot's signal, then corresponding video segments are classified as being contingent. Therefore, the contingent set included responses that were not covered by the game semantics that occurred when the robot did not say "Simon says." All video segments were coded independently by two of the authors, and for each video segment that was agreed upon, the coded time was averaged. The coder agreement was 100% for classification. (The average difference in coded time was 123 milliseconds.) We collected 246 video segments: 101 (97 contingent) cases from negotiation phases and 163 (120 contingent) cases from game phases.

8.1 Model Training

To train a computational model for contingency detection, we set or learn values for system parameters: time windows (W_B, W_A, and MRD) and conditional probability distributions for cues.

For the visual cues, we set the time window W_B and W_A to 4 seconds and 2 seconds, respectively. We chose these values because in the data, human responses usually last less than two seconds, and from our empirical observations, W_B should be at least two times longer than W_A to make a reliable dissimilarity evaluation. We learned the model of $P(S|C)$ from dissimilarity scores, which were obtained by processing contingent video segments from the game phase interactions in the training data set. Instead of manually collecting dissimilarity scores in which a response occurred, we extract one second of dissimilarity scores around the maximum.

A probabilistic model of $P(S|\overline{C})$, the null hypothesis is not learned from the training set since the null hypothesis should represent the amount of behavioral change that occurs naturally during interaction — not indicative of a contingent response. Therefore, it is learned on the fly by evaluating a human's normal behavior before the referent event. In building a null hypothesis, our proposed method requires both W_B+W_A and an extra α, used to generate dissimilarity samples. We set α to 2 seconds. Overall, the method requires 8 seconds of data.

For the audio cue, modeled as an event, we set the time window W_A for the audio cue to 200 ms. This is based upon our observation about how long the vocal responses take in our data set. We learn conditional models of $P(A_{on}|C)$ and $P(A_{on}|\overline{C})$ and a prior model of $P(C)$. These models are learned from both negotiation phase and game phase interactions in the training data set. $P(A_{on}|C)$ is simply calculated as the ratio of the total amount of time where audio events are found to be onset to the total amount of time for contingent interactions. We learned $P(A_{on}|\overline{C})$ in a similar manner using only non-contingent interactions. We set the prior model of $P(C)$ to 0.5 to eliminate classification bias towards contingency due to our unbalanced training data.

Depending on the choice of referent event, t_{Ref}, we use different evaluation windows MRD. When t_S is used as the referent event, the evaluation terminates after the following interaction begins or after it lasts for 8 seconds, whichever comes first. When MNI is used, MRD is set to 4 seconds, during which most of the responses happened in our "Simon says" game data.

8.2 Experimental Setup

Our experiments test the effects of cue integration and referent event selection on the overall performance of contingency detection in our experimental scenario. We

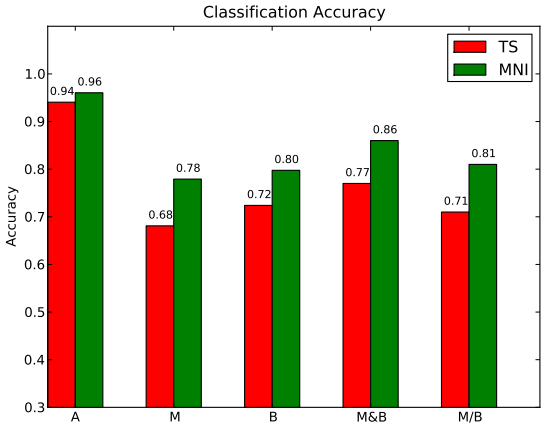


Fig. 14 Effects of referent event selection on accuracy of cues.

test 9 different cue combinations[1] and two types of referent events. One referent event is the time when the robot starts sending a signal, t_S. The other is the point of MNI in the robot signal.

To test effects of referent event selection, we build two different classifiers: TS (t_S as the referent event) and MNI (MNI as the referent event).

9 Results

9.1 Effects of Referent Event

As shown in Figure 14, the detectors with MNI as their referent event found visual responses in game phases and vocal responses in negotiation phases better than those with t_S. This reflects the fact that the closer the data for building a null hypothesis is to the referent event, the better the classification accuracy. When we extract data to model a human's background behavior, we want this data to be as new as possible and not to include any contingent responses. We find that the point of time when MNI happens meets these criteria for a good referent event.

The speed of exchanges in turn-taking interactions may not always permit the delay needed for building the null hypothesis. This problem occurred frequently in our collected data because the teleoperator focused on generating natural interactions with human-like timings rather than generating data for a contingency detector. Thus, many interaction segments do not contain enough data to model the human's baseline behavior. Using MNI as a referent event provides the detector a much tighter evaluation window than using the start time of the signal generated. We measured what percentage of data that is used for building null hypothesis comes from previous interactions. From our coded interactions, we observed that 67% of data (5.37 seconds) is from previous interactions in TS based detectors, 51% of data (4.15 seconds) in MNI based detectors.

In cases where the robot knows the MNI (i.e. from the gesture or speech that is being delivered), it is the best point in time to use as the referent. By using MNI, the robot knows when to stop evaluation. But in cases where this is not known, then the beginning of robot signal can serve as the fallback referent. The MNI is critical to making our contingency detectors operate in natural interactions in which the robot interacts with humans with human-like timings. This is necessary for realtime detection in realistic and natural interaction scenarios.

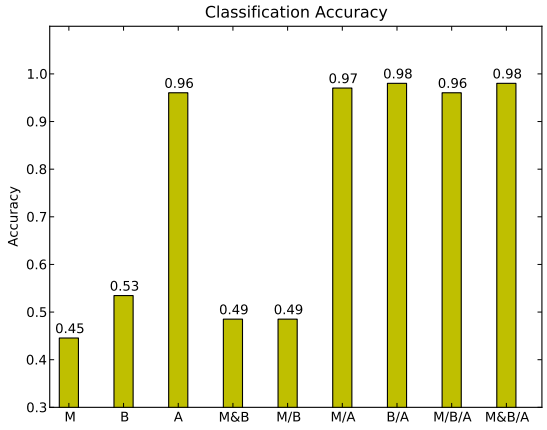
9.2 Cue Integration

We run experiments to test how indicative single or combined cues are for detecting the presence of responses. For the purpose of clarification, we explain our observations only with MNI-based classifiers. It should note that the same observations are valid with TS-based classifiers.

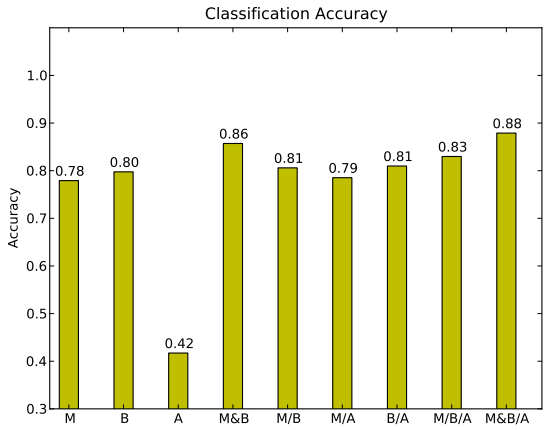
During negotiation phases, as shown in Figure 15(a), classifiers using audio cues detect speech-based responses with an accuracy of 0.96. The error of 0.04 is all false negatives and comes from interactions in which two robot actions occur in close sequence without delay. Thus, human subjects consider the two actions as one and wait to respond until the second action ends. Classifiers that do not use the audio cue perform under the average accuracy of 0.50. When the visual cues are merged together with the audio cue, the overall accuracy is slightly increased. This increase in accuracy results from more interaction data classified as being contingent; in some negotiation interaction cases, humans made visible responses by waving back to a robot without making vocal responses.

On the other hand, during game phases, the visual cues such as the motion, body pose, and the merged ones are more indicative of responses than the audio cue, as shown in Figure 15(b). Due to the nature of the game phase interactions, where a robot asks human subjects to imitate its motion commands, the most frequently detectable form of response is a visual change.

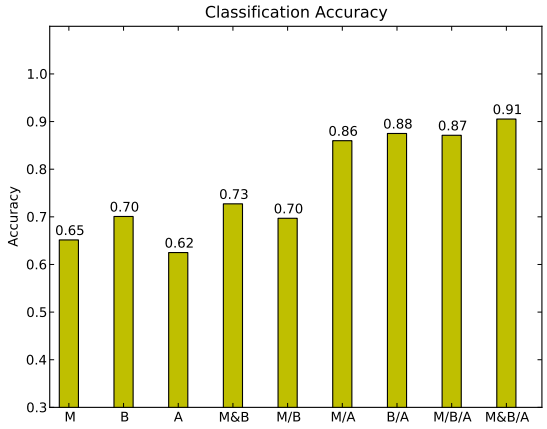
[1] M (Motion only), B (Body Pose only), A (Audio only), M&B (Motion and Body Pose merged at module level), M/B (Motion and Body Pose merged at decision level), M/A (Motion and Audio at decision level), B/A (Body Pose and Audio at decision level), M/B/A (all Motion, Body Pose, and Audio at decision level), M&B/A (Motion and Body Pose at module and Audio at decision level)



(a) Accuracy on the negotiation phase.



(b) Accuracy on the game phase.



(c) Accuracy on both the negotiation and game phases.

Fig. 15 Accuracy of the MNI based classifiers on both the negotiation and game phases. (MNI for their referent event) X&Y indicates cue X and cue Y combined at the module level; X/Y indicates integration at the decision level.

We test four different combinations for visual cues: 1) motion only, 2) body pose only, 3) motion and body pose merged at the module level, and 4) motion and body pose merged at the decision level. The best accuracy is obtained when the motion and body pose cues are integrated at the module level.

As shown, the accuracy of the classifier built with visual cues merged at the decision level (81%) is only 1% greater than that achieved by the best single visual cue of body pose. However, when integrated at the module level, the combination of pose and motion is 5% higher than pose alone. We argue that because visual cues model different but related aspects of the same perceptual signal, the cues are strongly related, and thus not conditionally independent. By combining at the module level this dependence is explicitly modeled and thus the merged cues generate more discriminative features. This is supported by our result that the classifiers built with a single visual cue or built with visual cues integrated at the decision level produce more false negative cases than the classifiers built with visual cues merged at the module level.

As shown in Figure 15(b), when the audio cue is merged, the overall accuracy is slightly increased. It is observed that when the audio cue is used together with the visual cues, more interaction data are classified as being contingent; sometimes, human subjects make some sound as a response during interactions, regardless of the robot's commands. For MNI data, there are 33 cases where human subjects make some sound while they are responding by imitating the robot and there are 10 cases where human subjects make sound when they are not supposed to imitate. These 10 cases are detected only with the audio cue.

When considering both negotiation and game phases, classifiers using either the visual cue or the audio cue perform well only for one phase, but not for the other. As shown in Figure 15(c), the average classification accuracy when using M, B, A, M&B, and M/B cues are less than 0.73. Overall, the best classification result, an accuracy of 0.91, is obtained when the motion and body pose cue are integrated at the module level and the audio cue is merged with integrated visual cues at the decision level.

10 Discussion

10.1 Three-Level Cue Integration

We integrated three different cues to improve contingency detection. There are many factors to be considered when determining the appropriate level for cue integration. They include whether to model a response as

an event or a change, and characteristics of cues (sampling rate and dimensionality of cues, the correlation between cue observations, and so on). The right integration level of the cue that is modeled as an event is the decision level because the presence of this event can be detected independent to the output of change-based detectors.

The best integration level, however, for cues used in change-based detectors depends on the correlation between cue observations. With our data, we obtain the best classifier when our visual cues are integrated at the module level and the audio cue is integrated at the decision level. Since the audio cue differs from the motion and body pose cues in terms of the sampling rate and dimensionality, it is integrated at the decision level. It is modeled as an event because some direct functional mapping from cue to response exists.

Visual cues are modeled as changes because there is no such direct mapping. At the decision level, the decision from each cue is merged with others assuming conditional independence. On the other hand, at the module level, the correlation between cue observations can be explicitly modeled in making a decision. The visual cues model the different aspects of the same perceptual region, so there exists heavy correlation between cues and thus one would predict an advantage for the module level of integration. Our experimental data support this conclusion.

10.2 Alternative to MNI as a Referent Event

We examined the effect of the point of MNI as a referent event on accuracy of contingency detection. The MNI point is the time at which a robot has finished conveying enough information for a human to give a semantically appropriate response. To detect a response, our contingency detector evaluates 4 seconds of data after the point of MNI. If the MNI point is not available, a default choice for the referent event would be t_S, the time at which the robot starts a signal. The size of the evaluation window with this referent event is uncertain and larger because the relative location of the MNI point with regard to t_S changes, so the MNI point could be possibly at one of many points between t_S and the time at which a robot finishes its signal.

We think that a more informative reference event and an evaluation window could be learned using contingency through interaction. A robot initially uses t_S as a referent event and a long timing window for evaluation. Every time a response is detected, the MNI is inversely estimated and is associated with each particular type of a signal made by the robot, whether it was speech-only, motion-only, or both. After obtaining enough associations, a robot adjusts its timing model and estimates the MNI without modeling information transfer from the start.

10.3 The Role of Contingency Detection in Interaction

We introduced a computational model for semantics-free contingency detection in which an event-based detector using the audio cue looks for the presence of vocal utterances and a changed-based detector using the visual cues looks for a presence of visible changes in human behavior. Because these cues model a generic property of a human's natural response, our learned model should be transferable to a new task.

We applied contingency detection in different interaction scenarios. In [3], we used contingency detection as a direct measurement of engagement. In order for a robot to draw a human's attention, a robot generates a probe signal to a human of interest and looks for some behavioral change. The presence of a contingent response from a human is a good indication that he is willing to have a follow-up interaction with the robot. We make a similar argument for detecting disengagement. When a human decides to finish interactions, if the robot fails to observe contingent responses, it is a good indication that a human has finished interacting with the robot.

In reciprocal turn-taking situations such as interactions from the "Simon says" game, contingency can be used as a signal. When taking turns with a human, a robot needs to know when to relinquish the floor. Contingency can be used as a signal for this. If the robot detects a human's contingent response, then it may indicate the human is ready to take a turn or already has it. The robot can then decide to yield the floor.

10.4 Limitation of Semantics-free Contingency Detection

The main limitation of our semantics-free contingency module is that detected responses are not grounded. In order for the robot to act appropriately on such responses, the robot might need additional information about responses such as semantics of responses and validity of responses with respect to the robot action (signal) and the context that the robot is situated in.

To overcome this limitation, we believe that a robot should model and recognize a set of grounded responses that are built from knowledge about the nature of the interaction situation, and should also be able to ground responses that are found by semantics-free contingency detection. Within the current framework, we can model

and recognize grounded responses as events. As future work, we will investigate how to attribute semantics to ungrounded responses through iterative interactions.

11 Conclusion

In this paper, we proposed a contingency detection framework that integrates data from multiple cues. We collected multimodal sensor data from a turn-taking human-robot interaction scenario based on the turn-taking imitation game "Simon says." We implemented our multiple-cue approach and evaluated it using motion and body pose cues from a structured light depth sensor and audio cues from a microphone. The results show that integrating multiple cues at appropriate levels offers an improvement over individual cues for contingency detection. We showed that using MNI as a referent event provides contingency detectors a more reliable evaluation window. From our constrained experiment, we made important observation that human response timings and effective integration of cues are important factors to detect contingency. We believe that this observation is important to understand other factors situated in more complex and ambiguous interactions. We believe that our contingency detection module improves a social robot's ability to engage in multimodal interactions with humans when the semantics of the human's behavior are not known to the robot.

References

1. K. Lohan, A. Vollmer, J. Fritsch, K. Rohlfing, and B. Wrede, "Which ostensive stimuli can be used for a robot to detect and maintain tutoring situations?" in *ACII Workshop*, 2009.
2. K. Pitsch, H. Kuzuoka, Y. Suzuki, L. Sussenbach, P. Luff, and C. Heath, "The first five seconds: Contingent stepwise entry into an interaction as a means to secure sustained engagement," in *IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, 2009.
3. J. Lee, J. Kiser, A. Bobick, and A. Thomaz, "Vision-based contingency detection," in *ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2011.
4. C. Chao, J. Lee, M. Begum, and A. Thomaz, "Simon plays simon says: The timing of turn-taking in an imitation game," in *IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, 2011.
5. H. Sumioka, Y. Yoshikawa, and M. Asada, "Reproducing interaction contingency toward open-ended development of social actions: Case study on joint attention," in *IEEE Transactions on Autonomous Mental Development*, 2010.
6. J. Triesch, C. Teuscher, G. Deak, and E. Carlson, "Gaze following: why (not) learn it?" in *Developmental Science*, 2006.
7. N. Butko and J. Movellan, "Infomax control of eye movements," in *IEEE Transactions on Autonomous Mental Development*, 2010.
8. G. Csibra and G. Gergely, "Social learning and social cognition: The case for pedagogy," in *Processes of Changes in Brain and Cognitive Development. Attention and Performance*. Oxford University Press, 2006.
9. K. Gold and B. Scassellati, "Learning acceptable windows of contingency," in *Connection Science*, 2006.
10. J. Watson, "Smiling, cooling, and 'the game'"," in *MerrillPalmer Quarterly*, 1972.
11. ——, "The perception of contingency as a determinant of social responsiveness," in *Origins of the Infant's Social Responsiveness*, 1979.
12. K. Gold and B. Scassellati, "Using probabilistic reasoning over time to self-recognize," in *Robotics and Autonomous Systems*, 2009.
13. A. Stoytchev, "Self-detection in robots: a method based on detecting temporal contingencies," in *Robotica*. Cambridge University Press, 2011.
14. B. Multu, T. Shiwa, T. Ishiguro, and N. Hagita, "Footing in human-robot conversations: how robots might shape participant roles using gaze cues," in *ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2009.
15. C. Rich, B. Ponsler, A. Holroyd, and C. Sidner, "Recognizing engagement in human-robot interaction," in *ACM International Conference on Human-Robot Interaction (HRI)*, 2010.
16. M. Michalowski, S. Sabanovic, and R. Simmons, "A spatial model of engagement for a social robot," in *International Workshop on Advanced Motion Control (AMC)*, 2006.
17. S. Muller, S. Hellbach, E. Schaffernicht, A. Ober, A. Scheidig, and H. Gross, "Whom to talk to? estimating user interest from movement trajectories," in *IEEE International Symposium on Robot and Human Interactive Communication (ROMAN)*, 2008.
18. N. Butko and J. Movellan, "Detecting contingencies: An infomax approach," in *IEEE Transactions on Neural Networks*, 2010.
19. D. Hall and J. Llinas, "An introduction to multisensor data fusion," in *Proceedings of the IEEE*, 1997.
20. R. Poppe, "A survey on vision-based human action recognition," in *Image and Vision Computing*, 2010.
21. The OpenNI API, http://www.openni.org.
22. M. Werlberger, W. Trobin, T. Pock, A. Wedel, D. Cremers, and H. Bishof, "Anisotropic huber-l1 optical flow," in *Proceedings of the British Machine Vision Conference(BMVC)*, 2009.
23. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *In Advances in Neural Information Processing*, 2000.

Jinhan Lee obtained his M.S. in Computer Science from the Georgia Institute of Technology in 2007. He is currently pursuing his Ph.D. degree in Robotics at the Georgia Institute of Technology. His research interests include perception for human-robot interaction, cue learning in multimodal interactions, and robot learning from demonstration.

Crystal Chao received her B.S. in Computer Science from the Massachusetts Institute of Technology in 2008. She is currently a Ph.D. student in Robotics at

the Georgia Institute of Technology. Her research interests include interactive learning, human-robot dialogue, and turn-taking in multimodal interactions.

Aaron F. Bobick is Professor and Chair of the School of Interactive Computing in the College of Computing at the Georgia Institute of Technology. He has B.Sc. degrees from MIT in Mathematics (1981) and Computer Science (1981) and a Ph.D. from MIT in Cognitive Science (1987). He joined the MIT Media Laboratory faculty in 1992 where he led the Media Lab DARPA VSAM project. In 1999 Prof. Bobick joined the faculty at Georgia Tech where he became the Director of the GVU Center and in 2005 became the founding chair of the School of Interactive Computing. He is a pioneer in the area of action recognition by computer vision, having authored over 80 papers in this area alone. His current focus is on robots understanding the affordances of objects and the action-based intentions of human beings.

Andrea L. Thomaz is an Assistant Professor of Interactive Computing at the Georgia Institute of Technology. She directs the Socially Intelligent Machines lab which is affiliated with the Robotics and Intelligent Machines (RIM) Center and with the Graphics Visualization and Usability (GVU) Center. She earned Sc.M. (2002) and Ph.D. (2006) degrees from MIT. She has published in the areas of Artificial Intelligence, Robotics, Human-Robot Interaction, and Human-Computer Interaction.